Assignment 2

Submission Date: 20/09/2025

1. Find the maximum element in an array. Solve this problem using **iterative** and **recursion** method

   Testcase1:
   Input: [10, 25, 47, 3, 19]
   Expected Output: 47

   Testcase2:
   Input: [-5, -10, -3, -20, -7]
   Expected Output: -3

2. Find the minimum element in an array.

   Testcases1:

   Input: [15, 8, 22, 5, 19]
   Expected Output: 5

   Testcase2:

   Input: [-4, -15, -7, -2, -30]
   Expected Output: -30

3. Calculate the sum of all array elements. Solve this problem using **iterative** and **recursion** method
   Testcase1:
   Input: [1, 2, 3, 4, 5]
   Expected Output: 15

   Testcase2:
   Input: [-1, 2, -3, 4, -5]
   Expected Output: -3

4. Find the average of array elements
   Testcase1:
   Input: [10, 20, 30, 40, 50]
   Expected Output: 30.0

   Testcase2:
   Input: [-5, 10, 15, -10, 5]
   Expected Output: 3.0
5. Print array elements in reverse order.
   Testcase1:

Input: [1, 2, 3, 4, 5]
Expected Output: [5, 4, 3, 2, 1]

Testcase2:
Input: [-1, 2, -3, 4, -5]
Expected Output: [-5, 4, -3, 2, -1]

6. Count even and odd elements in an array.
Testcase1:
Input: [1, 2, 3, 4, 5, 6]
Expected Output: Even: 3, Odd: 3

Input: [2, 4, 6, 8]
Expected Output: Even: 4, Odd: 0

7. Search for an element in the array (linear search).
Testcase1:
Input: [10, 20, 30, 40, 50], Search Element: 30
Expected Output: Element found at index 2

8. Copy elements of one array into another.
Testcase:
Input: Source Array: [1, 2, 3, 4, 5]
Expected Output: Destination Array: [1, 2, 3, 4, 5]

9. Display duplicate elements from an array.
Testcase1:
Input: [1, 2, 3, 4, 2, 5, 1]
Expected Output: 1, 2

Testcase2:
Input: [10, 20, 30, 40, 50]
Expected Output: No duplicates found

10. Find the second largest element in the array.

Testcase:
Input: [10, 20, 30, 40, 50]
Expected Output: 40

11. Create a LinkedList and insert elements at the end.

Testcase:
Existing LinkedList: [5, 10, 15]
Elements to insert: [20, 25]
Expected Output: LinkedList: 5 → 10 → 15 → 20 → 25

12. Insert a new node at the beginning of a LinkedList.

Testcase:
Existing LinkedList: [10, 20, 30]
Node to insert: 5
Expected Output: LinkedList: 5 → 10 → 20 → 30

13. Insert a new node at a given position in a LinkedList.

Testcase:
Existing LinkedList: [10, 20, 30, 40]
Node to insert: 25 at position 2
Expected Output: LinkedList: 10 → 20 → 25 → 30 → 40

14. Delete the first node of a LinkedList

Testcase:
Existing LinkedList: [10, 20, 30, 40]
Expected Output: LinkedList: 20 → 30 → 40

15. Delete the last node of a LinkedList.

Testcase:
Existing LinkedList: [10, 20, 30, 40]
Expected Output: LinkedList: 10 → 20 → 30

16. Delete a node by its value in a LinkedList.

Testcase:
Existing LinkedList: [10, 20, 30, 40]
Node to delete: 30
Expected Output: LinkedList: 10 → 20 → 40

17. Search for an element in a LinkedList.

Testcase:

Existing LinkedList: [10, 20, 30, 40]

Element to search: 30

Expected Output: Element found at index 2

18. Count the total number of nodes in a LinkedList.

Testcase:

Existing LinkedList: [10, 20, 30, 40]

Expected Output: Total nodes: 4

19. Reverse a LinkedList.

Testcase:

Existing LinkedList: [10, 20, 30, 40]

Expected Output: LinkedList: 40 → 30 → 20 → 10

Existing LinkedList: []

Expected Output: LinkedList: (empty)